ЗШ

УДК 004.934.1’1

***В.Ю. Шелепов, А.В. Ниценко***
Институт информатики и искусственного интеллекта ДонНТУ(Украина),
Институт проблем искусственного интеллекта НАН и МОН Украины
Украина, 83048, м. Донецк, ул. Артема, 118-б

# К проблеме распознавания слитной речи

***V.Ju. Shelepov, A.V. Nicenko***
*Institute of Informatics and Artificial Intelligence DonNTU (Ukrain),*
*Institute of Artificial Intelligence MES of Ukraine and MAS of Ukraine, c. Donetsk*
Ukraine, 83048, c. Donetsk, Artema st., 118-b

# *On the Problem of Continuous Speech Recognition*

***В.Ю. Шелєпов, А.В. Ніценко***
Інститут інформатики й штучного інтелекту ДонНТУ (Україна),
Інститут проблем штучного інтелекту МОН України і НАН України, м. Донецьк
Україна, 83048, м. Донецьк, вул. Артема, 118-б

# До проблеми розпізнавання злитої мови

В статье описывается предлагаемый авторами метод распознавания слитно произносимых фраз, состоящих из слов заданного словаря. Метод основан на использовании разработанного авторами механизма сегментации речевого сигнала и алгоритме нахождения первого слова, который авторы обозначают как «принцип минимума DTW-расстояния». Используется ранее предложенная авторами методика построения эталонов слов из дифонов и некоторая модификация метода DTW, дающая заметный выигрыш в скорости и объеме необходимой памяти. Практическая реализация метода требует также уточнения алгоритмов определения границ записанного речевого отрезка и расширения множества правил автоматического транскрибирования. Этому посвящены два первых раздела статьи.
**Ключевые слова:** начало и конец речи, транскриптор, слитная речь, сегментация, дифон, алгоритм DTW, выделение первого слова, принцип минимума DTW-расстояния.

Author’s recognition of continuous speech method is describing in the article. Phrase is consisting of words of given vocabulary. The method is based on using author’s mechanism of segmentation speech signal and algorithm of searching the first word, which we name *minimal DTW-distant principle.* We use earlier proposed by authors procedure of construction patterns with the help of diphone base and some modification of method DTW which gives win in speed and memory. Practical realization of method demands of more precise determination of speech boundaries and enlargement quantity rules of automatic transcription system. This is the subject of two first sections of the article.
**Key words:** beginning and end of speech, transcription system, continuous speech, segmentation, diphone, DTW-algorithm, the first word separation, minimal DTW-distant principle.

У статті описується пропонований авторами метод розпізнавання злито вимовлених фраз, що складаються зі слів заданого словника. Метод заснований на використанні розробленого авторами механізму сегментації мовного сигналу й алгоритму знаходження першого слова, що автори позначають як «принцип мінімуму DTW-відстані». Використовується раніше запропонована авторами методика побудови еталонів слів з дифонів і деяка модифікація методу DTW, що дає помітний виграш у швидкості й обсязі необхідної пам’яті. Практична реалізація методу вимагає також уточнення алгоритмів визначення границь записаного мовного відрізка й розширення множини правил автоматичного транскрибування. Цьому присвячені два перших розділи статті.
**Ключові слова:** початок і кінець мови, транскриптор, злита мова, сегментація, дифон, алгоритм DTW, виділення першого слова, принцип мінімуму DTW-відстані.

## 1 Видоизмененный алгоритм определения начала и конца речевого отрезка

Описываемый ниже алгоритм продолжает тему работы [1] и ориентирован на снижение влияния шума микрофона и звуковой карты.

Используется 8-битная запись с частотой 22050 Гц. По нажатии кнопки записи записываются последовательные отрезки звука по 300 отсчетов (окна). Для каждого из них вычисляется отношение $V/C$, где $V=\sum_{i=0}^{298}\left|x_{i+1}-x_i\right|$ – численный аналог полной вариации, $C$ – количество точек постоянства, то есть таких моментов времени, что в следующий момент величина сигнала остается той же самой. Берется среднее этого отношения по первым 10 окнам. Назовем эту величину «текущий StartPorog». Она характеризует верхний порог «молчания». Ждем момента, когда этот порог будет превышен не менее 5 раз подряд. Возвращаемся на 20 окон назад (начальный запас) и, начиная с этого момента, заносим записываемые отсчеты в буфер 1. Тем самым начинается запись того, что мы предполагаем речью. Определим «текущий EndPorog» как пятикратный текущий StartPorog. Заполнение буфера 1 продолжается до момента, после которого величины $V/C$ на протяжении 10 тысяч отсчетов будут меньше, чем текущий EndPorog. В него заносятся также упомянутые 10 тысяч отсчетов (запас в конце). Таким образом, запись предполагаемого речевого отрезка останавливается. Отметим, что при каждой записи вычисляются новые значения величин «текущий Start Porog» и «текущий EndPorog».

Записанное проверяется на наличие речи с использованием квазипериодичности ([2]). Если наличие речи обнаруживается, содержимое буфера 1 передается в буфер 2.

Записанный речевой отрезок сегментируется ([3]). Ввиду сказанного выше, сегментация будет начинаться и заканчиваться отрезком паузы (маркировка символом *P*). Наличие этого отрезка в конце позволяет определять, предшествует ли ему гласный (*W*) или звонкий согласный (*C*). Если заключительному *P*-отрезку непосредственно предшествует шипящий звук (*F*), алгоритм сегментации также позволяет его обнаружить.

Шум звуковой карты и микрофона может искажать информацию о границах речи. В связи с этим производится уточнение левой границы речевого отрезка. Для этого все записанное подвергается 100-кратному сглаживанию. При этом начальный отрезок молчания превращается в функцию времени, близкую к постоянной (значение этой постоянной определяется величиной первого отсчета, записанного в буфер 2). Считаем, что речь начинается с момента, когда отклонение от этой постоянной превышает порог $p_1$ (у нас это 10). Отмечаем этот момент в сигнале с помощью метки. Столь сильное сглаживание может «обрезать» начальный шипящий или часть звонкого согласного. Поэтому, если сегментация, произведенная выше, обнаруживает в начале отрезок шипящего или звонкого согласного, метка начала речи при необходимости сдвигается влево, в положение начала шипящего или согласного. Символ *P* в начале записи убирается.

Аналогичным образом производится уточнение правой границы записанного речевого отрезка.

Определяется наличие или отсутствие в конце речи глухого взрывного звука (*П*, *К*, *Т* или их мягкие варианты). Для этого подсчитывается расстояние (количество отсчетов) между последней меткой *P* и уточненной меткой конца сигнала. Если оно превышает некоторый порог $p_2$ (у нас это 2500), то считаем что в конце речи есть глухой взрывной и оставляем заключительный отрезок с маркировкой *P* у его левой границы. Если это расстояние меньше $p_2$, то заключительный *P*-отрезок убирается вместе с маркировкой метки *P* и эта метка считается истинным концом сигнала.

Пример 1

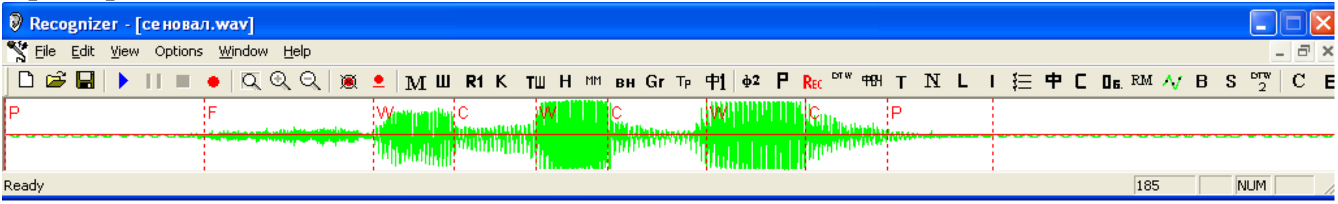



Рисунок 1 – Результат предварительной записи слова «Сеновал» с сегментацией

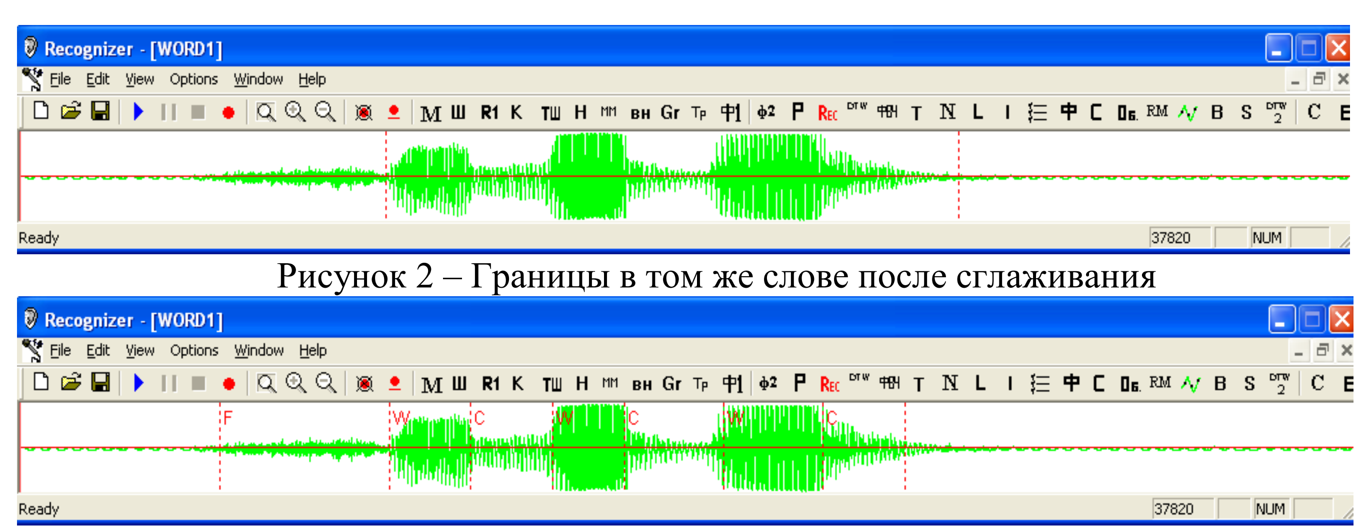



Рисунок 2 – Границы в том же слове после сглаживания

Рисунок 3 – Окончательный результат записи и сегментации слова «Сеновал»

Пример 2

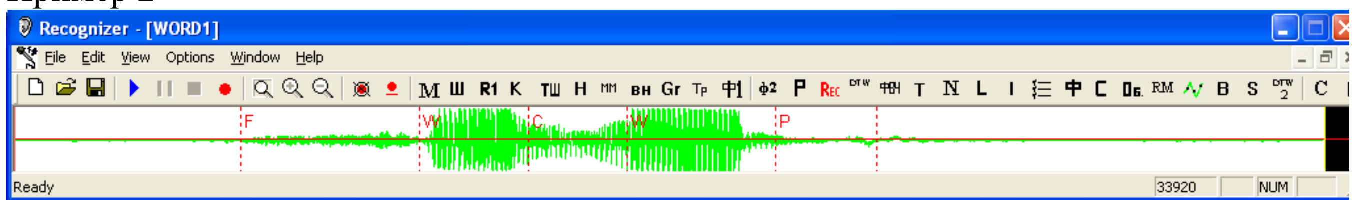



Рисунок 4 – Окончательный результат записи и сегментации слова «Салат»

Отметим в заключение, что использование упомянутой выше проверки на наличие речи позволяет организовать механизм автоматической записи, когда программа, записав слитный речевой отрезок, ожидает и записывает следующий, и пользователю нет необходимости каждый раз нажимать кнопку записи.

# 2 Расширенный транскриптор

Вновь появившиеся в связи с использованием дифонов возможности для различения звонких взрывных между собой, такие же возможности относительно глухих взрывных, твердых и мягких звуков, а также возникновение новых фонетических ситуаций на стыке слов при распознавании слитной речи, потребовало существенного расширения множества правил, заложенных в нашем автоматическом транскрипторе.

В качестве транскрипционных знаков для гласных звуков использованы в основном соответствующие русские буквы. Исключение составляют символы *w*, *q* для ударных *Е*, *Я* соответственно. Об особенностях этой ситуации сказано ниже. Твердые русские согласные транскрибируются также русскими буквами, а соответствующие мягкие согласные – аналогичными латинскими буквами. Исключения: значком @ обозначается мягкое *П*, значком $– мягкое *Ж*, значком & – южнорусское (украинское) *Г*, значком + обозначается слитный звук д'ж' (звонкая параллель Ч), значком % – слитный звук дз (звонкая параллель Ц).

Несколько предварительных слов о принципах, которых мы придерживаемся. Прежде всего, специалистам по распознаванию образов известно, что, как правило, увеличение числа классов распознавания ведет к снижению надежности распознавателя. Поэтому мы сознательно отказываемся от попыток тонкого распознавания аллофонов гласных фонем и используем для них единые транскрипционные символы *А, И, О, У, Э*, за одним исключением. В русском письме в том случае, когда за мягким согласным следует гласный, мягкость отражается путем выбора буквы для гласного: например, *А* после твердого *Д* в слове «да» и *Я* после мягкого *Д* в слове «дядя». Фонетически эти случаи отличаются очень сильно. Поэтому мы сохраняем в качестве транскрипционных знаков буквы *Е, Ё, Ю, Я*.

Транскриптор реализован как программа, заменяющая одни символы другими в соответствии с правилами, содержащимися в управляющем файле.

Вот его содержание на сегодняшний день:

1) =~

2) ~е=je, ~ё=jё, ~ю=jю, ~я=jя, ~\е=j\е, ~\ё=j\ё, ~\ю=j\ю, ~\я=j\я

3) ~=

4) "=
.=
,=
?=
!=

5) ого#=ова, его#=ева,

6) \асч=\ащ, \исч=\ищ, \осчи=\ощи, в\есч=в\ещ, досч\и=дощ\и, исч\ез=ищ\ез, исчез=ищез, насч=нащ, обсч=общ, пересч\и=перещ\и, пересчи=перещи, песч=пещ, пр\осч=пр\ощ, расч\ёс=расч\ёс=ращ\ёс, сч\ас=щ\ас, счас=щас, сч\ат=щ\ат, счето=щето, сч\ёт=щ\ёт, сч\ит=сч\ит=щ\ит, тсчит=тщит, счит=счит=щит, тысч=тыщ,

7) легк=лехк, лёгк=лёхк, м\ягк=мяхк,

8) здн=зн, дц=ц, тц=цц, тч=чч, жч=щ, зж=жж, сж=жж, сш=шш, стс=сс, стьс=сц, ндс=нс, нтс=нс, стн=сн, стц=сц, здц=сц,

9) \о=1, о=а, 1=\о, \е=w, \я=q,

10) ъе=je, ъё=jё, ъю=jю, ъя=jя, ьа=jя, ье=je, ьё=jё, ьэ=je, ью=jю, ья=jя, ъw=jw, ъ\ё=j\ё, ъ\ю=j\ю, ъq=jq, ьw=jw, ь\ё=j\ё, ь\о=j\ё, ь\ю=j\ю, ьq=jq, йа=йя, йу=йю, \а=йq,

11) ае=аje, аю=аjю, ая=аjя, ее=еje, ею=еjю, ея=еjя, ёе=ёje, ёю=ёjю, ёя=ёjя, ие=иje, ию=иjю, ия=иjя, ое=оje, ою=оjю, оя=оjя, уе=уje, ую=уjю, уя=уjя, ые=ыje, ыю=ыjю, ыя=ыjя, эе=эje, эю=эjю, эя=эjя, юе=юje, юю=юjю, юя=юjя, яе=яje, яю=яjю, яя=яjя, we=wje, wю=wjю, wя=wjя, qe=qje, qю=qjю, qя=qjя, аw=аjw, а\ё=аj\ё, а\ю=аj\ю, аq=аjq, еw=еjw, е\ё=еj\ё, е\ю=еj\ю, еq=еjq, ёw=ёjw, ё\ё=ёj\ё, ё\ю=ёj\ю, ёq=ёjq, иw=иjw, и\ё=иj\ё, и\ю=иj\ю, иq=иjq, оw=аjw, о\ё=аj\ё, о\ю=аj\ю, оq=аjq, уw=уjw, у\ё=уj\ё, у\ю=уj\ю, уq=уjq, ыw=ыjw, ы\ё=ыj\ё, ы\ю=ыj\ю, ыq=ыjq, эw=эjw, э\ё=эj\ё, э\ю=ыj\ю, эq=эjq, юw=юjw, ю\ё=юj\ё, ю\ю=юj\ю, юq=юjq, яw=яjw, я\ё=яj\ё, я\ю=яj\ю, яq=яjq, ww=wjw, w\ё=wj\ё, w\ю=wj\ю, wq=wjq, qw=qjw, q\ё=qj\ё, q\ю=qj\ю, qq=qjq,

12) #е=je, #ё=jё, #ю=jю, #я=jя, #w=jw, #\ё=j\ё, #\ю=j\ю, #q=jq, ~е=~je, ~ё=~jё, ~ю=~jю, ~я=~jя, ~w=~jw, ~\ё=~j\ё, ~\ю=~j\ю, ~q=~jq, стьд=zd,

13) бь=b, бj=bj, бе=bе, бё=bё, би=bи, бю=bю, бя=bя, бq=bq, бw=bw, б\ё=b\ё, б\и=b\и, б\ю=b\ю, бb=bb, вь=v, вj=vj, ве=vе, ви=vи, вю=vю, вя=vя, вq=vq,

вw=vw, вё=vё, в\ё=v\ё, в\и=v\и, в\ю=v\ю, вv=vv, гь=g, гj=gj, ге=ge, ги=gи, гю=gю, гя=gя, гq=gq, гw=gw, гё=gё, г\ё=g\ё, г\и=g\и, г\ю=g\ю, гg=gg, дь=d, дj=dj, де=de, ди=dи, дю=dю, дя=dя, дq=dq, дw=dw, дё=dё, д\ё=d\ё, д\и=d\и, д\ю=d\ю, дd=dd, зь=z, зj=zj, зе=ze, зи=zи, зю=zю, зя=zя, зq=zq, зw=zw, зё=zё, з\ё=z\ё, з\и=z\и, з\ю=z\ю, зz=zz, кь=k, кj=kj, ке=ke, кё=kё, ки=kи, кю=kю, кя=kя, кq=kq, кw=kw, к\ё=k\ё, к\и=k\и, к\ю=k\ю, кk=kk, ль=l, лj=lj, ле=le, ли=lи, лю=lю, ля=lя, лq=lq, лw=lw, л\ё=l\ё, лё=lё, л\и=l\и, л\ю=l\ю, лl=ll, мь=m, мj=mj, ме=me, ми=mи, мю=mю, мя=mя, мq=mq, мw=mw, мё=m\ё, м\ё=m\ё, м\и=m\и, м\ю=m\ю, мm=mm, нь=n, нj=nj, не=ne, ни=nи, ню=nю, ня=nя, нq=nq, нw=nw, нё=nё, н\ё=n\ё, н\и=n\и, н\ю=n\ю, нd=nd, пь=@, пj=@j, пе=@e, пё=@ё, пи=@и, пю=@ю, пя=@я, пq=@q, пw=@w, п\ё=@\ё, п\и=@\и, п\ю=@\ю, п@=@@, рь=r, рj=rj, ре=re, рё=rё, ри=rи, рю=rю, ря=rя, рq=rq, рw=rw, рё=rё, р\ё=r\ё, р\и=r\и, р\ю=r\ю, рr=rr, сь=s, сj=sj, се=se, си=sи, сю=sю, ся=sя, сq=sq, сw=sw, сё=sё, с\ё=s\ё, с\и=s\и, с\ю=s\ю, сt=st, фь=f, фj=fj, фе=fe, фи=fи, фю=fю, фя=fя, фq=fq, фw=fw, фё=fё, ф\ё=f\ё, ф\и=f\и, ф\ю=f\ю, фf=ff, хj=hj, хе=he, хи=hи, хю=hю, хя=hя, хq=hq, хw=hw, хё=hё, х\ё=h\ё, х\и=h\и, х\ю=h\ю, хh=hh, ть=t, тj=tj, те=te, тё=tё, ти=tи, тю=tю, тя=tя, тq=tq, тw=tw, тё=tё, т\ё=t\ё, т\и=t\и, т\ю=t\ю, тt=tt, зd=zd, зl=zl,

14) лзn=л2n, рзn=р2n, зn=zn, 2=з, ннщ=nщ, нn=nn, нt=nt, нч=nч, нщ=nщ, ccl=ssl, cl=sl, ccn=ssn, cn=sn, лст=л3t, рст=р3t, сt=st, 3=c, cs=ss,

15) ь=

16) б#=п, в#=ф, г#=к, д#=т, ж#=ш, з#=с, b#=@, v#=f, d#=t, z#=s,

17) бк=пк, бп=пп, бс=пс, бт=пт, бф=пф, бх=пх, бц=пц, бш=пш, вк=фк, вп=фп, вс=фс, вт=фт, вф=фф, вх=фх, вц=фц, вш=фш, гк=кк, гп=кп, гс=кс, гт=кт, гф=кф, гх=кх, гц=кц, гш=кш, дк=тк, дп=тп, дс=тс, дт=тт, дф=тф, дх=тх, дц=тц, дш=тш, жк=шк, жп=шп, жс=шс, жт=шт, жф=шф, жх=шх, жц=шц, жш=шш, зк=ск, зп=сп, зс=сс, зт=ст, зф=сф, зх=сх, зц=сц, зш=шш, бk=пk, б@=@@, бs=пs, бt=пt, бf=пf, бh=пh, бч=пч, бщ=пщ, вk=фk, в@=ф@, вs=фs, вt=фt, вf=ff, вh=фh, вч=фч, вщ=фщ, гk=kk, г@=к@, гs=кs, гt=кt, гf=кf, гh=кh, гч=кч, гщ=кщ, дk=тk, зтk=стk, д@=т@, дs=тs, дt=tt, дf=тf, дh=тh, wздч=wщ, здч=щч, дч=тч, дщ=тщ, жk=шk, ж@=ш@, жs=шs, жt=шt, жf=шf, жh=шh, жч=шч, жщ=шщ, зk=сk, з@=с@, зs=ss, зt=сt, зf=сf, зh=сh, зч=сч, зщ=щщ, bк=@к, bп=@п, bс=@с, bт=@т, bф=@ф, bх=@х, bц=@ц, bш=@ш, vк=fк, vп=fп, vс=fс, vт=fт, vф=fф, vх=fх, vц=fц, vш=fш, dк=tк, dп=tп, dс=tс, dт=tт, dф=tф, dх=tх, dц=tц, dш=tш, zк=sк, zп=sп, zс=ss, zт=sт, zф=sф, zх=sх, zц=sц, zш=sш, bk=@k, b@=@@, bs=@s, bt=@t, bf=@f, bh=@h, bч=@ч, bщ=@щ, vk=fk, v@=f@, vs=fs, vt=ft, vf=ff, vh=fh, vч=fч, vщ=fщ, dk=tk, d@=t@, ds=ts, dt=tt, df=tf, dh=th, dч=tч, dщ=tщ, zk=sk, z@=s@, zs=ss, zt=st, zh=sh, zч=sч, zщ=щщ,

18) кб=гб, кг=гг, кд=гд, кж=гж, кз=гз, пб=бб, пг=бг, пд=бд, пж=бж, пз=бз, сб=зб, сг=зг, сд=зд, сж=жж, сз=зз, тб=дб, тг=дг, тд=дд, тж=дж, тз=дз, тb=дb, тg=дg, тd=dd, тz=дz, фб=вб, фг=вг, фд=вд, фж=вж, фз=вз, хб=&б, хг=&г, хд=&д, хж=&ж, хз=&з, цб=%б, цг=%г, цд=%д, цж=%ж, цз=%з, чб=+б, чг=+г, чд=+д, чж=+ж, чз=+з, шб=жб, шг=жг, шд=жд, шж=жж, шз=жз, щб=$б, щг=$г, щд=$д, щж=$ж, щз=$з,
кb=гb, кg=gg, кd=гd, кz=гz, пb=bb, пg=бg, пd=бd, пz=бz, сb=зb, сg=зg, сd=зd, сz=zz, фb=вb, фg=вg, фd=вd, фz=вz, хb=&b, хg=&g, хd=&d, хz=&z,цb=%b, цg=%g, цd=%d, цz=%z, шb=жb, шg=жg, шd=жd, шz=жz,
@б=bб, @г=bг, @д=bд, @ж=bж, @з=bз, sб=zб, sг=zг, sд=zд, sж=жж, sз=zз, tб=dб, tг=dг, tд=dд, tж=dж, tз=dз,

fb=vb, fg=vg, fd=vd, fж=vж, fz=vz, @b=bb, @g=bg, @d=bd, @z=bz, sb=zb, sg=zg, sd=zd, sz=zz, tb=db, tg=dg, td=dd, tz=dz, fb=vb, fg=vg, fd=vd, fz=vz, чb=+b, чg=+g, чd=+d, чz=+z, щb=$b, щg=$g, щd=$d, щz=$z,

19) же=жэ, жи=жы, жю=жу, жя=жа, жw=ж\э, жё=ж\о, ж\ё=ж\о, ж\и=ж\ы, ж\ю=ж\у, жq=ж\а, ше=шэ, ши=шы, шю=шу, шя=ша, шw=ш\э, ш\ё=ш\о, ш\и=ш\ы, ш\ю=ш\у, шq=ш\а, це=цэ, ци=цы, цю=цу, ця=ца, цw=ц\э, ц\ё=ц\о, ц\и=ц\ы, ц\ю=ц\у, цq=ц\а, ча=чя, чу=чю, чэ=че, ч\а=чq, ч\о=чё, ч\у=чю, ч\э=чw, ща=щя, щу=щю, щэ=ще, щ\а=щq, щ\о=щё, щ\у=щю, щ\э=щw,

20) лл#=л, мм#=м, нн#=н,

21) \=

Поясним приведенный перечень правил. Каждое из них записано в виде двух или более частей, соединенных знаком =. Если упомянутых частей две, слева стоят исходные символы буквенной записи слова, справа – символы которыми они заменяются в транскрипции. Значок \ означает ударение. Машина, транскрибируя слово, последовательно ищет вхождение левой части очередного правила, и если таковое обнаруживается, заменяет его правой частью. Если упомянутых частей больше двух, создается соответствующее число вариантов транскрипции: вариант, соответствующий второй части, вариант, соответствующий третьей части равенства, и т.д.

Для удобства читателя в данном тексте правила разбиты на группы, которые занумерованы. Рекомендуется внести в управляющий файл эти группы в порядке номеров, не меняя порядка правил в группах, поскольку порядок замен, очевидно, важен.

Правило первой группы введено исключительно для наглядности. Оно временно заменяет пробел знаком ~.

Вторая группа описывает произношение *Е, Ё, Ю, Я* после пробела (начало слова).

Правило третьей группы убирает значок ~, а вместе с ним из транскрипции слитной речи уходят пробелы между словами.

Четвертая группа удаляет из транскрипции знаки препинания.

Пятая группа описывает произношение окончаний в родительном падеже прилагательных типа «нового», «синего».

Шестая группа служит для транскрибирования сочетания «*СЧ*» в различных ситуациях. Сочетание *СЧ*, которое в слове «считать» от слова «счёт» звучит, как *Щ*, даёт *СЧ* в омониме, обозначающем чтение с какого-то носителя; это порождает два варианта транскрипции.

Седьмая группа предназначена для описания произношения в словах типа «легко».

Восьмая – отражает произносительную норму в словах типа «мужчина» и некоторые фонетические правила, связанные с непроизносимыми согласными.

Девятая группа служит для транскрибирования гласных *О, Е, Я,* когда они стоят в ударной позиции. Поскольку транскриптор работает по принципу замены, приходится предварительно переименовывать ударное *О,* а затем возвращать ему прежнее обозначение. Введение специальных обозначений для ударных *Е, Я* связано с тем, что только они имеют совершенно определенное произношение. В безударном варианте они произносятся различными носителями языка по-разному. Для так называемой «младшей нормы» (более молодое поколение москвичей) они ближе к *И,* у сибиряков и в сценической речи – ближе к *Е, Я.*

Десятая группа правил отражает фонетическую роль мягкого и твердого знаков перед *Е, Ё, Ю, Я*. Их наличие приводит при произношении к появлению согласного *j*.

Одиннадцатая группа отражает произношение сочетаний гласных с гласными *Е, Ё, Ю, Я.*

Двенадцатая группа описывает произношение *Е, Ё, Ю, Я*, если с них начинается произносимый слитный речевой отрезок (# – знак начала и конца; в транскрибируемом тексте его проставлять не надо).

Тринадцатая и четырнадцатая группы связаны с обозначением в русском письме мягкости согласных и нейтрализацией твердых и мягких фонем.

Пятнадцатая – удаление мягкого знака, который уже сыграл свою роль.

Шестнадцатая группа – оглушение звонкой согласной в конце произносимого слитного речевого отрезка.

Семнадцатая группа – оглушение звонкой согласной перед глухой взрывной, шипящей и аффрикатами *Ц,Ч*.

Восемнадцатая – озвончение глухих согласных перед звонкими согласными.

Девятнадцатая группа отражает влияние твердого и мягкого согласного на последующий гласный.

Двадцатая – особенность произнесения удвоенных согласных в конце слова.

Отметим, что мы опробовали ряд достаточно успешных методов автоматического определения ударения в слове. Решение до конца этой трудной проблемы, очевидно, сильно сократило бы число слов – кандидатов на распознавание. Наш транскриптор пока убирает знак ударения (группа 21), но, в расчете на его автоматическое определение в звучащем слове, делает это лишь в самом конце.

Отметим, что мы включили в приведенный перечень лишь те правила, которые обусловлены русской фонетикой, и оставили за его пределами некоторые правила, порождаемые особенностями нашей сегментации. Например, сонорные согласные на конце слова после глухих взрывных сегментируются как гласные звуки (идентификатор *W*). Эти дополнительные правила включаются в отдельный транскриптор, используемый нами для создания файлов слов с транскрипцией широкой фонетической классификации (*W, C, F, P*), которые в данной работе не используются. Отметим также, что ряд вышеприведенных сочетаний не встречается в отдельных словах, но встречается в слитной речи.

Наконец, прежде, чем транскрибировать по указанным правилам, компьютер обращается к файлу исключений, в котором описываются процедуры транскрибирования целых слов, например,

чт\о=што, ог\о=ого.

# 3 Алгоритм с DTW-эталонами, создаваемыми из дифонов. Использование таблицы расстояний. Дерево эталонов. Размеры DTW-матрицы

Мы применяем для распознавания ставший уже классическим алгоритм Т.К. Винцюка, известный под названием алгоритма DTW (его описание в [1], также [2]). При этом мы используем свои вектора признаков, связанных с относительными частотами длин полных колебаний на речевых отрезках в 368 отсчетов ([4]). Эталоны слов распознаваемого словаря формируются из эталонов дифонов, полная база которых в объеме приблизительно полуторных тысяч создается для каждого диктора заранее (2 – 3 часа работы, [4]). Отметим, что создание такой базы в дальнейшем избавляет пользователя от необходимости создавать какие-либо эталоны голосом.

Под дифоном, соответствующим межфонемному переходу внутри слова, будем понимать участок стандартной длины: 3 окна в 368 отсчетов слева от метки между звуками и 3 таких же окна справа от той же метки. Эталон дифона – набор 6-и соответствующих векторов. Кроме того, мы используем участок в 3 окна в начале слова и участок в 3 окна – в конце слова, условно называя их соответственно начальным и конечным полудифоном слова (переход от молчания к речи и наоборот). Все вектора, входящие

в эталоны дифонов, играют роль кодовых векторов и образуют кодовую книгу *B*. Все эталоны дифонов нумеруются, нумеруются также все кодовые вектора.

Каждое слово словаря автоматически транскрибируется, по транскрипции строится цепочка имен дифонов. Каждое из них заменяется эталоном соответствующего дифона. Полученная цепочка векторов образует эталон слова ([4]).

На самом деле мы не создаем и не храним перечень эталонов слов словаря в виде статического списка. Словарь эталонов слов реализуется в виде дерева дифонов, использование которого существенно ускоряет процесс распознавания. Дерево создается при первоначальной загрузке текстового словаря. Дифоны представлены в дереве своими номерами. Эталон каждого слова представляется в виде ветви этого дерева. Если несколько ветвей имеют общую часть, то вычисления, заполняющие соответствующую часть DTW-матрицы, выполняются только один раз.

Уровни дерева соответствуют позициям дифонов в слове. Каждый узел в рамках каждого уровня представляет собой номер дифона, находящегося в слове на соответствующей позиции. Вершины, соответствующие конечным дифонам слов, помечаются как концы соответствующих слов (в узле записывается порядковый номер соответствующего слова в словаре). Если узел не конечный, то записывается значение -1. Максимальная глубина дерева соответствует максимальной длине (выраженной в количестве дифонов) слова в словаре.

Процесс распознавания строится следующим образом. Распознаваемое слово автоматически сегментируется и затем подвергается так называемой межфонемной обработке: удаляются стационарные части составляющих звуков и оставляются лишь дифоны в окрестностях межзвуковых меток (межфонемные переходы). Затем создается представление слова в виде набора *N* векторов признаков и строится таблица *D* расстояний этих векторов до всех векторов кодовой книги *B*. Далее вычисляются *DTW*-расстояния рассматриваемого слова до всех эталонов слов путем рекурсивного обхода дерева эталонов «в глубину». Вначале просматриваем корень дерева, а затем спускаемся по ветви, пока не достигнем вершины, помеченной как конец слова. После того, как достигнут конец слова, возвращаемся назад вдоль пройденного пути, пока не найдем вершину, у которой есть еще не посещенный сосед, а затем двигаемся в новом обнаруженном направлении. Процесс оказывается завершенным, когда мы вернулись в корень дерева, а все примыкающие к нему вершины уже оказались посещенными.

При прохождении ветвей дерева, по номерам дифонов строится цепочка соответствующих им номеров векторов, образующих эталон слова. Двигаясь в глубину, добавляем в цепочку номера, соответствующие пройденным узлам, а при движении назад они удаляются из нее. Достигнув узла, являющегося концом очередного слова, вычисляем *DTW*-расстояние от построенной цепочки векторов (эталона данного слова) до цепочки векторов распознаваемого сигнала. При этом расстояния между векторами берутся из таблицы *D*. В процессе вычисления расстояний матрица *DTW* не переписывается полностью, а обновляются только столбцы, соответствующие новым кодовым векторам, номера которых добавлены в цепочку после возврата назад по окончании предыдущего этапа.

Таким образом, достигается очень значительный выигрыш как в скорости распознавания, так и в объеме необходимой памяти.

В заключение отметим, что мы работаем с квадратными *DTW*-матрицами переменного размера: если эталон слова содержит $a$ векторов, а распознаваемое слово содержит $b$ векторов, то мы строим *DTW*-матрицу размера $\sqrt{a^2 + b^2}$ (Г.В. Дорохина).

## 4 Основные принципы предлагаемого подхода к распознаванию слитной речи

Пусть у нас есть несколько слитно произнесенных фраз. Наша программа автоматически затранскрибирует их и создаст для каждой из них эталон из дифонов, игнорируя пробелы между словами. После этого их можно распознавать между собой теми же методами, что и отдельно произносимые слова. Но если рассматривать множество произвольных фраз, то их бесконечно много и, очевидно, следует добиваться их распознавания путем распознавания слов, из которых они состоят. Тогда основная сложность – выделение в речевом сигнале отрезков, отвечающих отдельным словам. Иначе говоря, мы должны научиться определять, где заканчивается одно слово и начинается другое. Предлагаемый ниже метод основан на использовании вышеупомянутой сегментации. Весь рассматриваемый речевой отрезок автоматически разбивается на сегменты, отвечающие отдельным звукам, и границы между словами следует искать среди конечного множества полученных границ между звуками.

Мы начинали с распознавания пар слитно произносимых слов. Распознавая отрезок от начала до первой метки, а затем от первой метки до конца, мы получали пару слов нашего словаря. Затем мы проводили распознавание от начала до второй метки и от второй метки до конца и так далее. Заключительным шагом было распознавание всего речевого отрезка от начала до конца как одного слова. В результате мы получали последовательность гипотетических пар слов (на последнем месте – одно слово). Для каждой из этих пар автоматически строился эталон и результатом распознавания объявлялась пара, до которой *DTW*-расстояние минимально. Этот алгоритм показал высокую надежность. Но он включал целый набор актов распознавания отдельных гипотетических слов и в результате оказывался довольно долго работающим. Попытка применить аналогичный алгоритм к распознаванию большего числа слитно произнесенных слов ведет к экспоненциальному росту числа распознаваний гипотетических слов, и от нее приходится отказаться.

Тогда мы стали, двигаясь от начала до очередной метки, выводить только последовательность гипотез для первого слова, но с указанием DTW-расстояния до каждой из них. Оказалось, что гипотеза, соответствующая истинному первому слову (и соответствующему истинному отрезку от начала) имеет указанное расстояние, близкое к минимальному. Для слитно произносимых числительных (без фонетических вложений, о которых ниже) результат оказывался точным. Итак, мы приходим к следующему «принципу минимума»: **ПО КРАЙНЕЙ МЕРЕ, ДЛЯ СЛОВАРЕЙ, УДОВЛЕТВОРЯЮЩИХ НЕКОТОРЫМ ОГРАНИЧЕНИЯМ, ПЕРВОЕ СЛОВО ОПРЕДЕЛЯЕТСЯ С ИСПОЛЬЗОВАНИЕМ МЕТОК ИЗ УСЛОВИЯ МИНИМУМА DTW-РАССТОЯНИЯ.**

Понятно, что для распознавания второго слова фразы следует применить описанный метод к части сигнала от конца первого слова до конца речевого отрезка и так далее.

Смысл этого принципа становится понятен, если вспомнить, что алгоритм DTW направлен на минимизацию расстояния сказанного слова до эталона того же слова. Остальные слова в полученном списке на самом деле не звучали и то, что их расстояния до соответствующих эталонов оказались больше, представляется естественным.

Об ограничениях, упомянутых выше. Эксперименты показывают, что к числу таких ограничений нужно отнести следующее. Словарь не должен содержать пар слов,

одно из которых совпадает с началом другого. Точнее, не должно быть таких пар слов, что транскрипция одного из них получается из транскрипции другого приписыванием в конце дополнительных транскрипционных символов. В противном случае, при произнесении более длинного слова такой пары, DTW-расстояние до слова с более короткой транскрипцией может быть меньше.

## Литература

1. Шелепов В.Ю. Новый подход к определению границ речевого сигнала. Проблемы конца сигнала / В.Ю. Шелепов, А.В. Ниценко / Речевые технологии. – Москва, 2012.
2. Шелепов. В.Ю. Лекции о распознавании речи / Шелепов В.Ю. – Донецк : ІПШІ Наука і освіта. – 2009. – 192 с.
3. Шелепов В.Ю. Построение системы голосового управления компьютером на примере задачи набора математических формул / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук // Искусственный интеллект. – 2010. – № 3. – С. 259-267.
4. Шелепов В.Ю. О распознавании речи на основе межфонемных переходов / В.Ю. Шелепов, А.В. Ниценко, Г.В. Дорохина // Искусственный интеллект. – 2012. – № 1. – С. 132-139.
5. Винцюк Т.К. Анализ, распознавание и интерпретация речевых сигналов / Винцюк Т.К. – Киев : Наук. Думка, 1987. – 262 с.

## Literatura

1. Shelepov V.Ju. Speech technology / V.Ju. Shelepov, A.V. Nicenko. – Rechevye technologii. – Moskow, 2012.
2. Shelepov V.Ju. Lectures on speech recognition / Shelepov V.Ju. – Donetsk : IAI «Nauka i osvita», 2009. – 192 S.
3. Shelepov V.Ju. Artificial intelegence / V.Ju. Shelepov, A.V. Nicenko, A.V. Zhuk // Iskusstvennyj intellect. – 2010. – №.3. – S. 259-267.
4. Shelepov V.Ju. Artificial intelegence / V.Ju. Shelepov, A.V. Nicenko, G.V. Dorohina // Iskusstvennyj intellect. – 2012. – № 1. – S. 132-139.
5. Vincjuk T.K. Analysis, recognition and interpretation of speech signals / Vincjuk T.K. – Kiev : Naukova dumka, 1987. – 262 S.

**_RESUME_**

***W.Ju. Shelepov, A.V. Nicenko***

## *For the Problem of Continuous Speech Rrecognition*

The first section goes on the subject of article [1]. It describes algorithm determination of speech boundaries more robust relatively noise microphone and sound card. The second section describes automatic transcription system, which contains many new rules of explosive consonants transcription, soft consonants transcription and phonetic situations in boundaries of words.

We apply for recognition some modification of DTW-algorithm, using own system of signs ([4]). We create base of diphones, which contain phone transitions, and build patterns of words pasting together patterns of these diphones. We keep patterns of words in the form of diphones numbers tree. Using this tree we obtain essential saving of speed and memory.

*The main principals of suggested arrangement continuous speech recognition*

1) boundaries of words ought to look for in finite set of marks between sounds (our segmentation builds this set automatically).

2) Let us recognize (as a word of our vocabulary) the part of signal from the beginning to the first mark, then from the beginning to the second mark and so on. Then at least for vocabulary, which satisfy restriction given below, the first word is determined following minimal DTW-distant condition. (minimal DTW-distant principle).

Restriction is the next: vocabulary must not contain so pairs of words, that transcription one of them is the beginning of transcription the other. Otherwise, when we speak more long word, the distant to more shot word may be less.

*Статья поступила в редакцию 06.07.2012.*